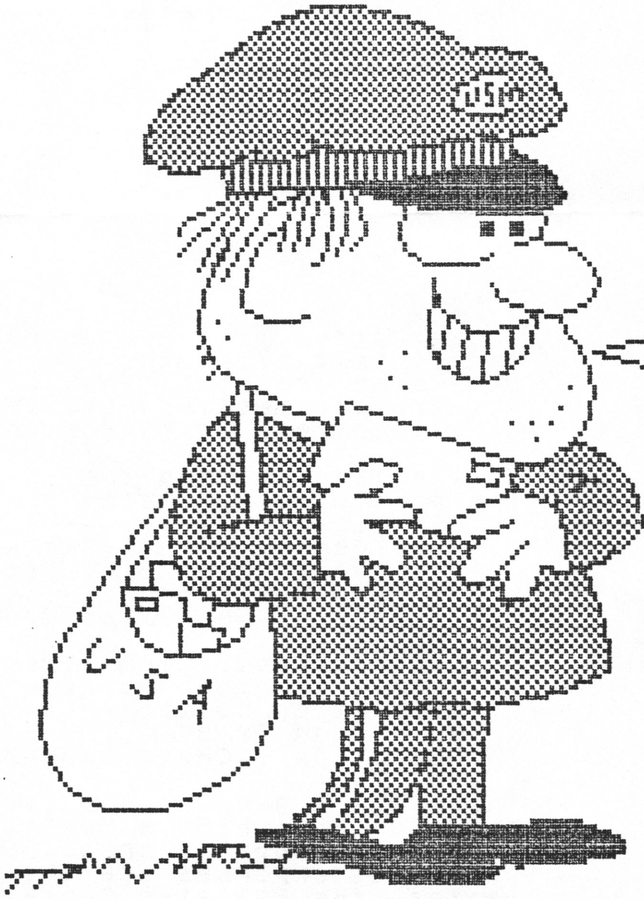
THE RAMTOP

SEPTEMBER **1985**



*** SPECIAL ***
MAILED MESSAGE!

All of you RAMTOP readers are experts in many computer fields. Please send in your articals so we can all share in your knowledge. Send to:
James G. DuPuy
6514 Bradley Ave. (DOWN)
Parma, Ohio 44129
PHONE: 216-661-4105

The TSU-BBS is ALIVE!!!!!!!!!
SYSOP - Chris Raynak
PHONE - 1-327-1099
UP-LOAD & DOWN-LOAD GAMES and
GENERAL T/S INFO.
PARAMETERS - 8/2/N
PASSWORD - sieg

If you forgot to renew your membership and subscription to THE RAMTOP! DO IT NOW !

<u>WEST SIDE</u> GROUP MEETS AT GETHSEMANE LUTHERAN CHURCH
14560 MADISON AVE. LAKEWOOD, OHIO 7:30 P.M.
EVERY THIRD FRIDAY EACH MONTH (EXCEPT DECEMBER)
CONTACT: DICK SIEG (216) 433-4387

<u>EAST SIDE</u> GROUP MEETS AT THE EUCLID SQUARE MALL
IN THE EUCLIDIAN ROOM 7:30 P.M.
EVERY FIRST FRIDAY EACH MONTH
CONTACT: MAX SCHOENFELD (216) 371-1096

A short note to our friends from other newsletters and magazines. You are welcome to use any of our material, news, adds, or programs if YOU: (1) Tell where it came from (RAMTOP Cleveland, Ohio) and (2) The author's name that wrote the article. We would appreciate it if you would send us a copy of the newsletter that it appeared in! Unless otherwise notified we will do the same.

THANK YOU FOR YOUR INTEREST IN OUR NEWSLETTER !

T/S RESOURCES

hardware software literature

August 1985 by Andy Kosiorek

Sinclair

Various UK magazines report that the relationship between Sir Clive and his new owner appear to be good. However there are many problems to be resolved. Example - Timex laid off 400 Spectrum Plus assembly workers because of poor sales.

Disk Drives

Both Aerco and Portugal Disk Drives are starting to reach actual users. Some users comments are; The Aerco system is good, but the DOS is not fully developed, and the instructions are very sparce, also the interface is uncased. The Portugese Drives DOS is excellent, but requires a lot of practice to use succesfully. The first shipments have the Spectrum buss configuration so they require a adapter board. If all goes well, I hope to be able to demonstrate a Drive at the Sept. meeting.

Modem Software

An excellant enhancement for the 2050 Modem software "MTERM-Smart II" has just been released. Titled "LOADER IV", the program features automatic redial of up to 20 phone numbers, auto loading of all numbers + system parameters, and the ability to load text files from Tasword II into the buffer. Documentation is good. At $7.95 this software is a best buy. Order direct from the author -

Kurt A. Casby
25 Battle Creek Court
St. Paul Mn. 55119

I have a correction from last month! On page 6 there is a reference to a magazine called the COMPUTER SMYTH. The address listed is WRONG! The error is that they are located in New Hampshire NOT New York. The full address is: PO BOX 176, Peter Borough, NH, 03458. Max Schoenfeld called to let me know of this.

CLUB NEWS

AUG. 29,1985

BOARD OF TRUSTESS FORMED:

At the Friday, Aug.17th west side meeting a Board of Trustees was formed to provide guidence and direction for the "club". The following members have agreed to serve as trustees.

Gregg DuPuy
Tom Jensen
Andy Kosiorek
Ted Kynszek
Bob Parrish
Chris Raynak
Dick Sieg
Tom Simon
John Velek
Gene Wilson

The trustees will meet monthly immediately after the regular west side group meeting. All trustee meetings are open to all members in good standing

At the first meetng the following persons were electd or appointed as officers;

President - Andy Kosiorek
Vice Pres. East Side Group- Gene Wilson
Vice Pres. West side Group- Dick Sieg
Corres. Sec. - Tom Simon
Treasurer - Bob Parrish
Newsletter Editor - Gregg DuPuy
BBS Sysop - Chris Raynak

There was a general discussion about establishing a Budget to cover the Clubs expenses. Our principal expenses are the news letter, the Bulletin Board, and meeting location rental fees. Our sources of income are membership dues, and hopefully, newsletter adds. This and other matters will be discussed at the next meeting on Sept. 21st. After three years of an informal but succesful existence, it is now up to the trustees to provide new directions and support for the Club.

Andy K. - pres.

Here is a FREE ad from one of our members. If you subcribe, you too may have a FREE ad!

IN THE BEGINING THERE WAS SINCLAIR
THEN THERE WAS BYTE-BACK

BYTE-BACK INC.

RT.4 BOX 54, LEESVILLE, SC 29070

BYTE-BACK HAS BEEN MANUFACTURING PERIPHERALS FOR TIMEX / SINCLAIR COMPUTER OWNERS SINCE 1981. IT HAS ADDED MANY NEW PRODUCTS SINCE ITS FIRST M16 AND BB-1 CONTROLLER. BYTE-BACK HAS PULLED THROUGH THE DEMISE OF TIMEX AND WILL CONTINUE TO PROVIDE SUPPORT AND NEW PRODUCTS IN THE FUTURE.

MODEM

MD-2B for ZX81,TS1000,TS1500
MD-68 for TS2068

ALL THE FEATURES NECESSARY FOR TELECOMMUNICATIONS WITH SERVICES SUCH AS COMPUSERVE OR YOUR LOCAL BBS

* upload and download text to and from memory and tape
* 300 baud - word length, parity, etc. selectable
* direct connect with phone load for clear communication
* RS-232 port for adding full size printers
* 2 hour compuserve demo pack
* 5 hour compuserve starter kit available separately

64K MEMORY

UM-64 for ZX81,TS1000
NOT AVAILABLE FOR TS2068

EXPAND YOUR TIMEX/SINCLAIR TO ITS FULLEST POTENTIAL

* 8-16K area selectable in 2K increments
* eprom socket for preprogrammed eproms (2K and 4K)
* reset switch which only resets the 16 to 32K area
* battery back up to stop annoying crashes
* rom-ram transfer for machine code buffs
* this is not a nonvolatile ram!

CONTROLLER

BB-1 for ZX81,TS1000,TS1500
BB-68 for TS2068

CONTROL THINGS WITH YOUR TIMEX/SINCLAIR

* 8 independent relays
* 8 led status indicators to monitor status of relays
* analog to digital converter sold separately to allow controller to measure voltages
* 8 logic inputs

RS-232

RS-232 for ZX81,TS1000,TS1500
RS-232-68 for TS2068

YOUR TIMEX SINCLAIR CAN OPERATE FULL SIZE RS-232 TYPE PRINTERS

* cable and software sold separately
* C.ITOH 7500AR serial printer available separately

PARALLEL

PARALLEL 1000 for ZX81,TS1000,TS1500
PARALLEL 2068 for TS2068

YOUR TIMEX SINCLAIR CAN OPERATE FULL SIZE CENTRONICS PARALLEL TYPE PRINTERS

* 5ft cable and software provided
* C.ITOH 7500AP parallel printer available separately

90 DAY WARRENTY ON ALL MODULES

Any hardware module may be returned for a full refund within 10 days of receipt.
BYTE-BACK offers a few hardware kits for kit builders. Our kits require excellent soldering ability.
Kits include a high quality printed circuit board and all of the parts.
Kits have a 90 day warrenty on parts only. Kits may not be returned for a refund once they have been started.

TELEPHONE (803) 532-5812

__UM-64 assembled	$119.95	__kit $109.95
__BB-1 assembled	$ 69.00	__kit $ 59.00
__BB-68 assembled	$ 69.00	__kit $ 59.00
__RS-232 assembled	$ 69.95	__kit $ 59.95
__RS-232-68 assembled	$ 69.95	__kit $ 59.95
__MD-2B assembled	$149.95	
__MD-68 assembled	$149.95	
__A-D converter assembled	$ 29.95	
__Test lead for A-D	$ 3.95	
__PARALLEL 1000 assembled	$ 84.95	
__PARALLEL 2068 assembled	$ 84.95	
__CABLE FOR RS-232	$ 19.95	
__PRINTER SOFTWARE FOR RS-232	$ 19.95	
__ZX PRO/FILE (TS1000)	$ 16.95	
__ZX PRO/FILE (TS2068)	$ 29.95	
__COMPUSERVE 5 HOUR PACKAGE	$ 29.95	
__7500AR SERIAL PRINTER	$329.00	
__7500AP PARALLEL PRINTER	$289.95	

THERE WILL BE A $4.95 SHIPPING CHARGE PER ORDER.
PLEASE SPECIFY COMPUTER TYPE ____________

PLEASE BILL MY __AM.EXP. __VISA __M/C

CARD #____________________ EXP DATE______

__ I HAVE ENCLOSED A CHECK OR MONEY ORDER

NAME____________________

ADDRESS____________________

____

CITY/STATE/ZIP____________________

PHONE____________

EDITORS NOTES FOR SEPTEMBER

Hi all! As you can see, our newsletter, the RAMTOP is getting bigger and better! I want you to please note that we now are including PAID ads in the RAMTOP! This will help us to grow even bigger and better! BUT don't just skip over these ads! We all should try our best to be LOYAL to our advertisers! They are important to us! Byte Back is a VERY reputable company and from the reviews I have seen, their products are well worth the money! Dave Hoshor feels that the 007-SPY is the best for copying programs for backup. He also says that it is better than File Copy 3. If you purchase from them, MAKE SURE THAT YOU TELL THEM YOU SAW IT IN THE RAMTOP!!!

As many of you know, we represented Timex/Sinclair at the Quaker Square computer fair. Tom Simon, Max Schoeenfeld, Clarence Lucht, and I were there Sat. and Chris Raynak and Dave Hoshor were there Sun. We handed out over 100 imformation sheets with imfo about our group, a couple of programs and a cut out to send for a free RAMTOP. We also had several people pay dues right there for a years subscription! WELCOME to our NEW members! We are dedicated to bringing you quality news, reviews, programs, and the latest that we can dig up! I want you 1000/1500, ZX-81 people to know that as long as there is one BIT of material for your systems, we will back you up! I learned most of my programing on the ZX-81. They will live on for a long time yet! I know that many of you 1000/1500 people have the word processor program in this issue, but Roy has a few changes and I'm sure that it may be of benefit.

I want ALL of you that contribute with your articles, that we all REALLY appriciate your efforts! Keep those articles coming! I future issues you will see more ads, programs, software updates, hardware projects, reviews, news, and LOTS more! *THANKS AGAIN FOR ALL YOUR HELP!!! TAKE CARE ALL!!!*

James G. DuPuy

SOFTWARE SOURCES NEWS

SPEEDYSOFT, formerly Software Supermarket has changed its address and phone number. The service is as good as ever. The new address is:

Speedysoft
37 Church Road
London, SW13 9HQ
England

tel. 01-846-9353

CURRY COMPUTER of Glendale, Arizona is a reliable source of Spectrum software, literature, and hardware. Their address is

Curry Computer
5344 West Banff Lane
Glendale Arizona 85306

tel. 602-978-2902

Among their offerings are English computer magazines at $4 each or 3 for $10.50. They also have Softaid, a group of ten Spectrum programs, the proceeds of which are sent to African famine relief, for only $9.95. Curry Computer's prices include shipping.

ZX GUARANTEED offers a Spectrum tape copier that will copy programs with pulsed headers or fast loaders. It won't copy every program, but it's the best copier I've seen to date. The program costs £7.00 or $15.00 by airmail.

ZX Guaranteed
29 Chadderton Dr.
Unsworth,
Bury,
Lancs. BL9 8NL
England

David Hoshor

Here is a GREAT article! I hope you will take the time to write to Eric and express your appreciation to him!!!

ON-LINE
with Eric Yruegas

BASIC PRIMER

Welcome to the **BASIC PRIMER**! This month, I am going to discuss using the PRINT statement with variables.

As you know, the PRINT statement is used to print things onto the screen. Last month, you learned how to print numbers and text messages. Now I'll show you a different way to do the same thing.

Before I do that, a quick review of VARIABLES is in order. A variable is a place in the computer's memory for temporary storage of data. In layman's terms, that means that there is a place in your computer to put things, so that you can get them back later when you need them.

There are two types of variables: **STRINGS**, which can hold anything (numbers or text), and **NUMERIC**, which is restricted to numbers only.

Using variables with the PRINT statement is very easy. All you must do is substitute the variable for the data to be printed. Let me give you an example:

```
PRINT "HELLO"
```

becomes

```
PRINT A$
```

NOTE: Before you can use the PRINT statement with A$, you must "set up" A$ beforehand, by using the LET statment. Example:

```
10 PRINT "HELLO"
```

becomes

```
10 LET A$="HELLO"
20 PRINT A$
```

That's all there is to it! There are a few tricks you can do with variables too. You can print more than one string at once! Look at this:

```
10 PRINT "HELLO THERE"
```

can become

```
10 LET A$="HELLO "
                ^--(space)
15 LET B$="THERE"
20 PRINT A$+B$
```

Neat, huh? You would get the same thing on the screen with either program. The same thing applies to the NUMERIC variables, too. Example:

```
10 PRINT 12345

   can become

10 LET A=12345
15 PRINT A
```

Simple enough? That same trick I used with the string variable still applies:

```
10 PRINT 100

   can become

10 LET A=90
15 LET B=10
20 PRINT A+B
```

You can substitute any two numbers that add up to 100 for the 90 and the 10 in lines 10 and 15. You can even combine strings and variables, if you so desire. Here's one way:

```
10 LET A$="MY NUMBER IS "
                         ^--(space)
15 LET A=25
20 PRINT A$+A
```

Type that one in your computer, and see what happens. This concludes the BASIC PRIMER for this month. Drop me a line (see address at the end of this column) if you need help, or if you have a question.

ADVANCED PROGRAMMING

Here we are again, in the Machine Code Corner. This month, I am going to show you some ROM routines that use the TS2040 printer! But first, an apology.

Last month, I gave you two routines that scrolled part of the screen left and right. For one reason or another, the BASIC loader did not get printed. For those who looked all over the newsletter for it, please forgive me. You will find it at the end of this month's column, along with a reprint of the HEX codes and mnemonics.

Now some of you, I am sure, know how to print things to the printer from machine code already. For those of you who do not, this month's discussion should help you along.

First, the PRINTER BUFFER. Located at 23296 decimal, 5B00 hex, it contains information to be dumped to the printer itself. The buffer is only 256 bytes long (FF hex), so the computer can send only so much at one time.

The buffer is set up just like the display file (oh no!), only it is one line of data. 32 bytes wide, and 8 bytes deep, exactly like one line in the display file. This allows you to send bits of information to the printer, instead of just character codes, like other printers.

Now we get to the good stuff. The printer dump routine itself is not that long, if you don't count the subroutines that it uses. It is located at 2595 decimal, or 0A23 hex. This routine simply takes what is in the printer buffer and sends it to the printer. Then it clears the buffer and returns to where it came from. For those of you who go through the function dispatcher for the printer routines, you no longer need to punish youself with such travesty! You can go straight to the source.

The COPY command from BASIC is just as easily done in machine code as it is to type COPY from the keyboard. The routine to do it is located at 2562 decimal, 0A02 hex. It sets up a loop (for the 22 lines of the screen to be copied), loads the printer buffer with the first 8 "scan" lines (when I say "scan" lines in this column, I mean the 8 horizontal bytes used to form a character on the screen) of the display file, and calls the printer dump routine. Then it counts down the loop, loading the buffer with the next line of data from the display file, dumping it, and so on and so forth until all 22 lines are copied. But what about the bottom two lines of the screen? There are a lot of programs that use those lines and how in the world can I get them to the printer? Read on, my friend...

There is a way to print all 24 lines of the screen to the printer, but it requires just a teensy bit of preparation to do so. Also, you need to call a different address in the ROM. Here is what you need to do. Load your hex loader or assembler, or if you don't have one, get out the BASIC loader below and type the routine in. After doing so, **PLEASE** (!!!) save it to tape. Here is the HEX codes with mnemonics, with the BASIC loader following:

24 LINE SCREEN DUMP

(THIS ROUTINE CAN BE LOCATED ANYWHERE IN MEMORY)

```
F3          DI          ; TAKE OUT INTERRUPTS
06C0        LD B,C0     ; LET B=192
CD050A      CALL 0A05   ; RANDOMIZE USR 2565
C9          RET         ; BACK TO CALLER

10 FOR N=65350 TO 65356: READ X: POKE N,X: NEXT N
15 DATA 243,6,192,205,5,10,201
```

There you have it. All 24 lines to the printer. You can change the 192 in the BASIC loader or the C0 in the hex listing, so that you could copy only 8 lines of the screen... Just change it to 64 decimal in the BASIC loader and in the hex listing, to 40 hex. Very simple!

If you want to print more or less than that, all you must do is change the B register to however many "scan" lines you want printed. Keep in mind that it counts from the **top** of the screen, so you can't copy the middle of the screen, unless you use yet another routine...

The next routine I am going to give you can be used to copy nearly any part of the screen, 32 columns across (I haven't worked out a way to print only 16 columns yet).

You must first find out the address of the first byte to copy (it can be anywhere in the display file), and then find out how many "scan" lines you want to print.

Then you call the ROM routine! Here it is:

PART-COPY ROUTINE

(THIS ROUTINE CAN BE LOCATED ANYWHERE IN MEMORY)

```
21NNNN     LD HL,NNNN      ; HL=START OF COPY
06NN       LD B,NN         ; B=# OF SCAN LINES
F3         DI              ; TAKE OUT INTERRUPTS
CD080A     CALL 0A08       ; RANDOMIZE USR 2568
C9         RET             ; BACK TO CALLER
```

```
10 FOR N=65350 TO 65359: READ X: POKE N,X: NEXT N
15 DATA 33,NN,NN,6,NN,243,205,8,10,201
```

NOTE: "NN" MEANS YOU PUT IN YOUR OWN NUMBERS OR ONES OFF OF THE FOLLOWING TABLE!

Here are some key spots on the screen:

HEX	DECIMAL	COMMENT:
4000	16384 (0,64)	AT 0,0
4020	16416 (32,64)	AT 1,0
4040	16448 (64,64)	AT 2,0
4060	16480 (96,64)	AT 3,0
4080	16512 (128,64)	AT 4,0
40A0	16544 (160,64)	AT 5,0
40C0	16576 (196,64)	AT 6,0
40E0	16608 (224,64)	AT 7,0
4800	18432 (0,72)	AT 8,0
4820	18464 (32,72)	AT 9,0
4840	18496 (64,72)	AT 10,0
4860	18528 (96,72)	AT 11,0
4880	18560 (128,72)	AT 12,0
48A0	18592 (160,72)	AT 13,0
48C0	18624 (196,72)	AT 14,0
48E0	18656 (224,72)	AT 15,0
5000	20480 (0,80)	AT 16,0
5020	20512 (32,80)	AT 17,0
5040	20544 (64,80)	AT 18,0
5060	20576 (96,80)	AT 19,0
5080	20608 (128,80)	AT 20,0
50A0	20640 (160,80)	AT 21,0
50C0	20672 (196,80)	AT 22,0 - INPUT AREA
50E0	20704 (224,80)	AT 23,0 - INPUT AREA

Well, there you have it. All you have to do is copy the numbers in parentheses () off of the table and into the BASIC loader. Then set up the B register with the number of scan lines, and call the routine!

That is just about it for this month. Send comments to:

ERIC YRUEGAS
4706 LANGLEY AVE.
WHITEHALL, OH 43213

Now, for that BASIC loader for SCREEN SCROLL:

```
          1 REM SCREEN SCROLL- USR 65280 FOR LEFT, USR 65298  FOR
     RIGHT.
          5 CLEAR 65279
         10 FOR N=65280 TO 65315: READ X: POKE N,X: NEXT N
         15 DATA 33,255,71,14,32,167,203,22,43,13,32,250,62,64,188,32,24
2,201
         20 DATA 33,0,64,14,32,167,203,30,35,13,32,250,62,72,188,32,242,
201
```

Here is the HEX dump:

```
                        TOP THIRD TO THE LEFT
FF00 21FF47        LD HL,47FF        ; HL=18431
FF03 0E20          LD C,20           ; C=32
FF05 A7            AND A             ; CLEAR CARRY
FF06 CB16          RL (HL)           ; SEE JULY NEWSLETTER
FF08 2B            DEC HL            ; HL=HL-1
FF09 0D            DEC C             ; C=C-1

FF0A 20FA          JR NZ,FF06        ; GOTO FF06 IF C<>0
FF0C 3E40          LD A,40           ; A=64
FF0E BC            CP H              ; IF A=H THEN SET Z FLAG
FF0F 20F2          JR NZ,FF03        ; GOTO FF03 IF H<>A
FF11 C9            RET               ; BACK TO CALLER

                        TOP THIRD TO THE RIGHT
FF12 210040        LD HL,4000        ; HL=16384
FF15 0E20          LD C,20           ; C=32
FF17 A7            AND A             ; RESET CARRY FLAG
FF18 CB1E          RR (HL)           ; SEE JULY NEWSLETTER
FF1A 23            INC HL            ; HL=HL+1
FF1B 0D            DEC C             ; C=C-1
FF1C 20FA          JR NZ,FF18        ; GOTO FF18 IF C<>0
FF1E 3E48          LD A,48           ; A=72
FF20 BC            CP H              ; IF A=H THEN SET Z FLAG
FF21 20F2          JR NZ,FF15        ; GOTO FF15 IF Z FLAG NOT SET
FF23 C9            RET               ; BACK TO CALLER
```

This is Eric's first article for us. I sure hope he will send us more! **THANKS ERIC!** I can see that you spent a lot of time on this one. It is very well thought out!

Here is another Very helpful article by Dave Hoshor. I'm sure that you will find it quite interesting!

Aug 19 20:51 1985   Page 1

TECHNIQUES FOR CALLING USR ROUTINES

If you've examined some of your programs for any of the Sinclair computers containing machine code, you've probably seen machine code routines called in a variety of ways. The most common calls are:

```
RANDOMIZE USR n     or
LET A = USR n
```

where "n" represents the address where the machine code is located.

How do you know which way to call a USR routine? The purpose of this article is to explain why USR routines are called in the ways that they are, and to suggest some alternate ways of calling USR routines.

THE BASICS

As you probably know, the heart of Sinclair computers (QL's excepted) is the Z80 microprocessor. When using BASIC in our programs, we are communicating with Z80 in an indirect way. Our commands are "interpreted" by the ROM into instructions that are meaningful to the Z80 microprocessor. The extra step of interpreting the BASIC program takes times and means that BASIC programs run slower. Sinclair computers allow us to execute programs that are directly executable by the Z80 microprocessor by use of the USR function. The USR function removes us from the safety of BASIC, since you can easily cause the computer to crash. Don't worry, you can't hurt the computer, but you can easily lose a program painstakingly entered into the computer. You are working without a net when you run machine code programs. The results are breathtaking when the program works, but until your program is fully debugged, you are only a step away from disaster. For this reason, SAVE your machine code programs before running them.

When invoking the USR function, we are telling the computer to execute a machine code program at a specific address. Inside the Z80 microprocessors, there are three "register pairs" that can hold any integer between 0 and 65535. These register pairs are called the HL, DE, and BC registers. (There are some other registers, but they are more difficult to use.) When we write machine code programs, we manipulate the contents of these registers, and the contents of RAM to some useful purpose.

WHAT HAPPENS ON RETURNING TO BASIC FROM A USR ROUTINE?

In most cases, after finishing a USR routine, we return to BASIC. On returning to BASIC, THE CONTENTS OF THE BC REGISTER PAIR ARE INTERPRETED AS A NUMBER. This fact allows us to pass information from our machine language routines back to BASIC.

Let's write a program that does nothing but load the BC register pair with the number 50 decimal, and return to BASIC. (This will work on any Sinclair computer.)

Enter:

```
POKE 30000,1
POKE 30001,50
POKE 30002,0
POKE 30003,201
```

To the Z80 microprocessor, this program means, load the BC register with 50, then return. In Z80 mnemonics this is abbreviated as:

```
                LD BC,32  (that's 50 in hexadecimal)
                RET
```

Now let's prove that the contents of the BC register are interpreted as a number on return to BASIC. Enter:

```
                PRINT USR 30000
```

The computer will print 50. Try entering:

```
                LET A=USR 30000
                PRINT A
```

Again the computer prints 50, but this time the contents of the BC register has been assigned to the variable A.

A common USR call is:

```
                RANDOMIZE USR n
```

The effect of this call is to put the contents of the BC register in the system variable SEED. SEED is used to provide a number for random numbers. To see the results of RANDOMIZE USR 30000 enter:

```
        TS 1000, TS 1500, ZX-81 version
                PRINT PEEK 16434 + PEEK 16435 * 256

        TS 2068, SPECTRUM version
                PRINT PEEK 23670 + PEEK 23671 * 256
```

Again, the computer prints 50, the number that has been stored in SEED. (You can find the system variables by looking in the appendix of your manual.)

You should be careful when using the RANDOMIZE USR n command. If you are using random numbers somewhere in your program as in a game, the RANDOMIZE USR n command may keep putting the same number in SEED time and time again. This will mean that you will keep getting the same random numbers again and again. After using RANDOMIZE USR n, you should have another line that says RANDOMIZE or RANDOMIZE 0. This will cause the number of TV frames shown since you turned your computer on to be deposited in SEED. This number should be fairly random.

In the preceding examples we have seen three common ways of calling USR routines. Now let's examine a couple more advanced methods.

ADVANCED USR CALL TECHNIQUES

Since we have seen that the BC register pair is interpreted as a number on return to BASIC, it is possible to use USR calls in any way that a numeric argument is valid. For example you could have a line that said:

```
                IF USR n = 10 THEN GOTO 1000
```

and it would be perfectly legal. You could execute multiple USR routines with a statement such as

```
LET A = (USR n OR USR m OR USR p)
```

and all three USR routines would be executed.

I enjoy storing data in strings. String storage of data has the advantage of being compact, and the Sinclair computers handle strings especially well. My favorite USR call is:

```
LET A$(1) = CHR$ USR n
```

It has the advantage of setting the system variable DEST to contain the address of A$(1). This means that you can store data in strings and easily find the location of that string for your machine code routine. (DEST will also point to numeric variables, but they are quite a bit more difficult to use.) The one thing you have to remember when using DEST is that you must restore a$(1) to some legal character code between 0 and 255. This is easily done by ending your routine with:

```
LD HL,(DEST)
LD C,(HL)
LD B,0
RET
```

This loads the BC register with the character that is pointed to by DEST. It is much like saying:

```
LET A$(1)=A$(1)
```

One final tip on calling USR routines. If you need to set a value to a USR routine from BASIC you can always POKE it to some unused location in memory, but here is a bit more elegant way. Pass the number to memory with the RANDOMIZE function. Suppose you want to set the number 3 to the routine. Just enter RANDOMIZE 3 and the number three will be put in SEED. Then call your USR routine and have it read SEED. It's simple, and will even put two byte numbers in SEED. The one number it won't work for is zero.

Once you have learned that the BC register is the key to returning to BASIC, you will find that your machine code routines will be more imaginative.

David Hoshor

On the next several pages we have a Word Processor for the TS-1000/1500 on ZX-81. This is a modifacation of the Word Juggler program that many of you have already. You may just be able to study this and modify the one you have. THANKS ROY!

****
****

WORD PROCESSOR FOR TS 1000--1500

BY ROY D. ZELESNIK 7/10/85

****
****

THIS IS A MODIFIED VERSION OF THE **WORD JUGGLER** PROGRAM LISTED IN A BOOK TITLED **THE SINCLAIR ZX81 - PROGRAMMING FOR REAL APPLICATIONS** BY RANDLE HURLEY. THE AUTHOR GIVES PROGRAMMING TIPS AND LISTINGS WITH FULL EXPLANATIONS OF SIX OTHER USEFUL PROGRAMS PLUS SOME HARDWARE MODIFICATIONS.

BEFORE KEYING IN THE PROGRAM READ ALL INSTRUCTIONS THOROUGHLY. YOU CAN SAVE TIME KEYING IN THE FAST MODE. THE PROGRAM ITSELF HAS BEEN USED TO PRINT THESE INSTRUCTIONS.

****

USER INSTRUCTIONS

****

TO WRITE TEXT

****

KEY 1 FOLLOWED BY ENTER TO SELECT (WRITE) MODE. WHEN YOU HAVE ENTERED 32 CHARACTERS THE SCREEN WILL BLANK FOR A SHORT WHILE BEFORE THE CURSOR APPEARS ON THE NEXT LINE. IT IS DURING THIS BLANK PERIOD THAT THE TEXT IS COPIED INTO THE MEMORY.

TO RETURN TO THE MAIN MENU

****

KEY (SHIFT)/Q. NOTE THAT YOU CANNOT RETURN TO THE MAIN MENU FROM THE (EDIT) MODE.

TO MOVE BETWEEN (EDIT) AND (WRITE) MODES

****

KEY (SHIFT)/M.

TO EDIT

****

USE THE CURSOR CONTROLS (ARROWS) BUT WITHOUT THE SHIFT KEY. NOTE THAT THE (UP) ARROW MOVES THE CURSOR UP A COMPLETE PARAGRAPH AT A TIME WHEREAS THE (DOWN) ARROW MOVES THE CURSOR DOWN ONE LINE AT A TIME. TO RETURN TO THE MENU FROM THE (EDIT) MODE, YOU MUST FIRST GO TO (WRITE) MODE BY KEYING (SHIFT)/M.

TO FILE TEXT ON TAPE

****

SELECT (FILE TEXT) FROM THE MENU BY KEYING 5 THEN ENTER. START THE RECORDER (SET IT TO RECORD) AND AFTER A FEW SECONDS TO ALLOW THE MOTOR SPEED TO STABILIZE, KEY ENTER AGAIN. THE PROGRAM WILL RETURN TO THE MENU WHEN RECORDING IS COMPLETE. STOP THE RECORDER. MAKE A BACKUP COPY IF YOU WISH BY REPEATING THE OPERATION.

TO PRINT TEXT

****

SELECT THIS OPTION FROM THE MENU BY KEYING 6 AND THEN ENTER. THE COMPUTER WILL ASK FOR THE NUMBER OF THE LAST PARAGRAPH TO BE PRINTED - EVERYTHING FROM THE BEGINNING OF THE TEXT TO THIS POINT WILL BE PRINTED OUT. ENTER THE NUMBER AND KEY ENTER. WHEN THE SCREEN DISPLAYS (FINISHED), KEY ENTER TO RETURN TO THE MENU.

TO READ TEXT

****

SELECT THIS OPTION FROM THE MENU BY KEYING 3 THEN ENTER. PRESS ENTER REPEATEDLY (OR HOLD IT DOWN) TO STEP THROUGH THE TEXT. AT THE END OF THE EXT THE PROGRAM WILL RETURN TO THE MENU.

TO DELETE PARAGRAPHS

****

SELECT THIS OPTION BY KEYING 4 AND THEN ENTER. YOU WILL BE ASKED WHICH PARAGRAPH YOU WANT TO DELETE. ENTER THE PARAGRAPH NUMBER, FOLLOWED BY ENTER. YOU HAVE TO CONFIRM THE DELETION BY KEYING (D) FOLLOWED BY ENTER. THE SCREEN WILL BLANK FOR A TIME YOU CAN DELETE SUCCESSIVE PARAGRAPHS BY KEYING (D) FOLLOWED BY ENTER AGAIN. REMEMBER TO RETURN TO THE MENU AT LEAST ONCE EVERY 18 PARAGRAPHS TO AVOID CRASHING THE PROGRAM WITH A SCREEN FULL OF MESSAGES.

TO RESTART THE PROGRAM

****

IF FOR ANY REASON YOU SHOULD CRASH THE WORD PORCESSING PROGRAM (E.G. BY KEYING -BREAK-) AND IT RETURNS TO THE LISTING, RESTART BY KEYING (GOTO 3000) FOLLOWED BY ENTER.

SPECIAL INSTRUCTIONS

THE PROGRAM ITSELF TAKES 3.5K OF MEMORY SO ONLY 12.5K ARE AVAILABLE FOR TEXT STORAGE. 57 OF THE 6 LINE PARAGRAPHS CAN BE HELD IN THE COMPUTER BEFORE MEMORY LIMITATIONS CAUSE TROUBLE. THIS IS EQUIVALENT TO AROUND 2000 WORDS, A GOOD SIZE ESSAY.

THE SPACE ABOVE LINE 220 CAN BE USED FOR THE MAIN PROGRAM BUT THE LINES BETWEEN 1 AND 100 MUST BE LEFT STRICTLY ALONE. THE PROGRAM ONLY WORKS BECAUSE THE ZX81 KNOWS EXACTLY WHICH MEMORY STORE HOLDS THE FIRST CHARACTER IN THE FIRST STRING. ONE CHARACTER BETWEEN LINE 1 AND 100 AND THE MACHINE WILL NO LONGER KNOW WHERE IT IS AND THE PROGRAM WILL CRASH.

NO QUOTATION MARKS MAY BE USED WHILE WRITING TEXT. USE ( ) INSTED. IF QUOTATION MARKS WERE TO BE POKED INTO A STRING THEN THE MACHINE WOULD INTERPRET THIS AS THE END OF THE STRING AND THE STRUCTURE OF THE STORAGE WOULD BE SPOILED. SO, YOU HAVE TO AVOID PRESSING (SHIFT)/P.

USE THE (ZERO) OR THE DELETE KEY AS THE SPACE KEY. DO NOT USE THE BREAK KEY OR YOU WILL HAVE TO RETURN TO THE MAIN MENU AND START OVER AGAIN. USE THE LETTER O KEY IF YOU NEED TO PRINT ZERO.

YOU WILL FIND TWO EXTRA PIECES OF HARDWARE VERY USEFUL; A PIECE OF BLACK TAPE TO SHOW WHERE THE END OF LINE 7 IS ON YOUR TV AND A PIECE OF PLASTICINE OR SIMILAR MATERIAL TO PUT ON THE BREAK KEY AS A REMINDER NOT TO USE IT TO WRITE A SPACE.

DEVELOP A RHYTHMIC STYLE OF TYPING AND KEY IN DOUBLE LETTERS AT THE SAME RATE THAT YOU KEY IN THE OTHERS. THE STRIP OF BLACK TAPE ON THE SCREEN SAVES A LOT OF CODE AND GIVES YOU A FULL 32 CHARACTERS TO THE LINE. THE END OF A LINE IS THE MOST AWKWARD PART OF THE WHOLE PACKAGE BECAUSE A MISTAKE THERE IS MORE DIFFICULT TO CORRECT. THE PROCEDURE IS:

1) GO INTO EDIT MODE.
2) KEY 7 FOR THE LAST PARAGRAPH
3) KEY 6 UNTI THE LINE WITH THE MISTAKE IS REACHED.
4) KEY 5 OR 8 UNTIL THE MISTAKE IS COVERED BY THE CURSOR.
5) GO BACK TO THE WRITE MODE AND CORRECT THE MISTAKE.

HITTING THE BREAK KEY BY MISTAKE WILL SOON BECOME A RARE EVENT BUT FOR THE MOMENT, HERE IS WHAT TO DO. THE SIGNAL IS THE REPORT CODE (D) BUT, BEING ENGROSSED IN YOUR WORK YOU WILL NOT NOTICE UNTIL THE PROGRAM REPLACES YOUR TEXT ON THE SCREEN. EITHER KEY GOTO 3000 OR KEY CONTINUE. GOTO 3000 WILL REQUIRE THAT YOU KNOW THE PARAGRAPH NUMBER. USING CONTINUE WILL MEAN HAVING TO GO INTO EDIT MODE BEFORE MOVING TO THE NEXT PARAGRAPH TO PRINT TEXT ON THE SCREEN. ONCE THE TEXT IS REPRINTED THE CURSOR CONTROL KEYS SOON MOVE YOU BACK TO THE PRINT POSITION.

EDITING IS ONLY PERMANENT IF THE LINE BEING EDITED IS COPIED INTO THE STORAGE STRINGS. CORRECTING THE MISTAKE AND THEN GOING INTO EDIT MODE AGAIN TO MOVE ON TO THE NEXT CORRECTION ONLY ALTERS THE TEXT ON THE SCREEN. AFTER MAKING A CORRECTION, USE THE CURSOR KEYS IN EDIT MODE TO MOVE TO THE END OF THE LINE OR, IF IT IS MORE CONVENIENT, TYPE OVER THE CHARACTERS AGAIN. WHEN AT THE END OF THE LINE, MAKE SURE YOU ARE IN THE WRITE MODE AND THEN TYPE THE LAST SPACE OR CHARACTER AND WAIT THE SECOND NEEDED TO COPY THE LINE. THE CORRECTION WILL NOW BE PERMANENT.
NOTE: ALSO REMEMBER THAT EACH LINE OF 32 CHARACTERS OF TEXT MUST BE KEYED IN FOR THE LINE TO BE STORED IN MEMORY. INCLUDE ALL SPACES AT THE END OF THE LAST SENTENCE IN A PARAGRAPH OR USE THE METHOD DESCRIBED ABOVE.

WHEN SETTING UP STORAGE YOU WILL HAVE TO KEY IN 192 SPACES, BUT YOU WILL ONLY HAVE TO DO IT ONCE THE OTHER 56 PARAGRAPHS CAN ALL BE SET UP USING THE PROGRAM EDIT MODE (SHIFT)/1.

KEY IN THESE LINES:

```
1   GOTO 3000

100 LET A$= (KEY IN 192
            SPACES - REMEM-
            BER TO USE QUO-
            TATION MARKS BE
            FORE AND AFTER)

101 PRINT LEN A$
```

AND THEN RUN THE PROGRAM TO MAKE SURE THE STRING IS EXACTLY 192 SPACES LONG. NEXT REPLACE LINE 101 WITH:

```
101 RETURN
```

NOW THAT ONE OF THE SUB-ROUTINES IS SET UP CORRECTLY USE THE EDIT (SHIFT)/1 KEY TO BRING DOWN LINE 100 AND CHANGE IT TO 102. THEN ADD LINE (103 RETURN) ETC. UNTIL YOU REACH LINE 213. YOU WILL NOW HAVE FIFTY SEVEN 6-LINE PARAGRAPHS. NEXT PROCEDE TO KEY IN THE ENTIRE PROGRAM. GOOD LUCK.

```
   1 GOTO 3000
 100 LET A$="






                   "
 101 RETURN
 102 LET A$="






                   "
 103 RETURN
 212 LET A$="






                   "
 213 RETURN
 900 LET A=198+PEEK (16396)+PEEK
 (16397)*256
 901 RETURN
 910 CLS
 911 IF P>57 THEN GOTO 2990
 912 FAST
 913 GOSUB P*2+98
 914 PRINT A$
 915 GOSUB P*2+100
 916 PRINT A$
 917 GOSUB P*2+102
 918 PRINT A$
 919 PRINT AT 19,0;"PARAGRAPH ";
P
 920 SLOW
 921 RETURN
 930 LET P=P+1
 931 GOSUB 910
 932 GOSUB 900
 933 RETURN
1000 LET P=0
1030 GOSUB 930
1040 LET C=1
1050 LET L=1
1070 POKE A+C,PEEK (A+C)+128
1080 IF INKEY$<>"" THEN GOTO 108
0
1090 IF INKEY$="" THEN GOTO 1090
1100 LET A$=INKEY$
1110 IF A$="""" THEN GOTO 2990
1120 IF CODE (A$)>63 THEN GOTO 1
080
1130 IF A$=">" THEN GOTO 1500
1140 IF A$="0" THEN LET A$=" "
1170 POKE A+C,CODE (A$)
1180 LET C=C+1
1190 IF C=33 THEN GOTO 1300
1200 GOTO 1070
1300 FAST
1310 LET C=1
1320 SCROLL
1330 FOR J=1 TO 32
1340 POKE 16501+J+L*32+P*209,PEE
K (A-33+J)
1350 NEXT J
1360 LET L=L+1
1370 IF L=7 THEN GOSUB 930
1380 IF L=7 THEN LET L=1
1390 SLOW
1400 GOTO 1070
1500 PRINT AT 20,16;"CURSOR"
1510 IF INKEY$<>"" THEN GOTO 151
0
1520 IF INKEY$="" THEN GOTO 1520
1530 LET A$=INKEY$
1540 PRINT AT 20,16;"          "
1550 GOSUB 900
1560 POKE A+C,PEEK (A+C)-128
1570 IF A$="5" THEN LET C=C-1
1580 IF A$="8" THEN LET C=C+1
1590 IF A$="9" THEN LET C=C+5
1600 IF C<1 OR C>32 THEN LET C=1
1610 IF A$="N" THEN LET L=1
1620 IF A$="N" THEN GOSUB 930
1630 IF A$="7" THEN LET P=P-2
1640 IF A$="7" THEN LET L=1
1650 IF A$="7" THEN GOSUB 930
1660 IF A$="6" THEN SCROLL
1670 IF A$="6" THEN LET L=L+1
1680 IF L=7 THEN GOSUB 930
1690 IF L=7 THEN LET L=1
1700 POKE A+C,PEEK (A+C)+128
1710 IF A$=">" THEN GOTO 1080
1720 GOTO 1500
2990 CLS
3000 PRINT TAB 7;"WORD PROCESSOR
"
3010 PRINT
3020 PRINT "KEY 1 TO WRITE TEXT"
3030 PRINT "KEY 2 TO EDIT"
3040 PRINT "KEY 3 TO READ"
3050 PRINT "KEY 4 TO DELETE A PA
RAGRAPH"
3060 PRINT "KEY 5 TO FILE TEXT"
3070 PRINT "KEY 6 TO PRINT TEXT"
3100 INPUT A$
3110 IF A$="1" THEN GOTO 1000
3120 IF A$="2" THEN GOTO 3200
3130 IF A$="3" THEN GOTO 3500
3140 IF A$="4" THEN GOTO 4000
3150 IF A$="5" THEN GOTO 5000
3160 IF A$="6" THEN GOTO 4500
3180 GOTO 2990
3200 CLS
3210 PRINT TAB 10;"EDITING"
3220 PRINT AT 5,0;"KEY PARAGRAPH
 NO."
3230 INPUT A
3240 LET P=A-1
3250 LET A$="N"
3260 LET C=1
3270 GOTO 1610
3500 LET P=2
3510 GOSUB 910
3520 IF INKEY$="" THEN GOTO 3520
3530 LET P=P+3
3540 GOTO 3510
4000 CLS
4010 PRINT TAB 10;"DELETING"
4020 PRINT
4030 PRINT "WHICH PRAGRAPH?"
4040 INPUT P
4050 PRINT "KEY D TO DELETE PARA
GRAPH ";P
4060 INPUT A$
4070 IF A$<>"D" THEN GOTO 2990
4075 FAST
4080 FOR J=1 TO 192
4090 POKE 16533+J+P*209,0
4100 NEXT J
4105 SLOW
4110 LET P=P+1
4120 GOTO 4050
4500 CLS
4510 PRINT "KEY THE LAST PARAGRA
PH TO BE"
4515 PRINT "PRINTED"
4520 INPUT A
4530 CLS
4540 PRINT AT 10,10;"PRINTING"
4550 FOR J=1 TO A
4560 GOSUB J*2+100
4570 LPRINT A$
4580 NEXT J
4600 PRINT AT 10,10;"FINISHED"
4610 INPUT A$
4620 GOTO 2990
5000 CLS
5010 PRINT TAB 9;"START TAPE";TA
B 9;"PRESS ENTER"
5020 INPUT A$
5030 CLS
5040 LET A$="WORD PROCESSOR"
5050 SAVE A$
5060 GOTO 3000
6000 REM WORD PROCESSOR
```

I got a letter from Jack Roberts of T/S Connections last month. He wanted to set us straight about his opperation. I will thus pass this on to you. 1- T/S Connections is NOT now nor have they ever been an authorized service center for Timex. 2- Timex Products Service Dept. in Little Rock Ark. still does In and OUT of warranty service. 3- Users have the option of where to send their OUT of warranty units to.

I understand from Ted Knyszek who wrote to them, that they will sell SOME parts for the 2068 or 1000 and 1500s. From what Ted told me, I feel that you should inquire to Timex and/or T/S Connections for cost of repair before purchasing parts due to there high costs. T/S Connections has a LIMITED supply of parts so they are always hunting for new supplies. T/S Connections also has several products to enhance your computer. These are: A power switch for the 2040 printer, Stereo jack for the 2068, LED for the 2068. The Greater Cincinnati Users Group has a 2068 based BBS that is now operational and being improved daily. It will be operational nation wide with the addition of the Aerco Disk Drive System. Unfortunately, they didn't include the phone number but I will give it to you as soon as I get it. If you wish to enquire to T/S Connections for a repair, parts, or products, write to: The T/S Connection, 3832 watterson Ave., Cincinnati, OH., 45227.

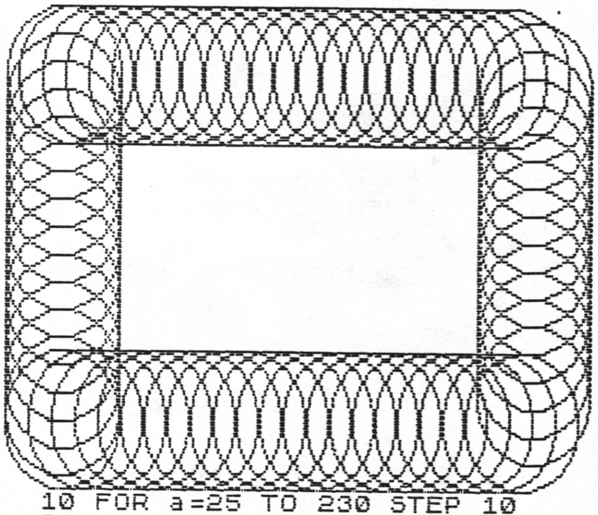
Here is a program that will set up the graphics in the quotes. This is from William Walker. His address is: 6226 Walmot Dr., Huntington, WV 25705

Thanks William!

THESE PROGRAMS FOR THE 2068 OR SPECTRUM.

```
   10 REM Sampler for programers
   ******************************
   20 REM W Simister*************
******************************
   30 REM This POKE makes the key
s BEEP *************************
   40 POKE 23609,255: BORDER 5: I
NK 0
 200 REM Making single graphics
  ******************************
 205 FOR x=0 TO 7: READ y: POKE
USR "a"+x,y: NEXT x: REM A Cocke
rel
 210 DATA 48,255,49,62,62,28,20,
34
 215 FOR x=0 TO 7: READ y: POKE
USR "b"+x,y: NEXT x: REM B Eleph
ant
 220 DATA 0,48,56,126,95,159,18,
18
 225 FOR x=0 TO 7: READ y: POKE
USR "c"+x,y: NEXT x: REM C Eleph
ant
 230 DATA 0,12,28,126,250,249,72
,72
 235 FOR x=0 TO 7: READ y: POKE
USR "d"+x,y: NEXT x: REM D Tree
 240 DATA 16,56,56,124,124,254,1
6,16
 245 FOR x=0 TO 7: READ y: POKE
USR "e"+x,y: NEXT x: REM E Deer
 250 DATA 16,32,64,224,127,63,50
,82
 255 FOR X=0 TO 7: READ y: POKE
USR "h"+x,y: NEXT x: REM H
 260 DATA 4,72,223,62,8,0,0,0
 265 FOR x=0 TO 7: READ y: POKE
USR "i"+x,y: NEXT x: REM I
 270 DATA 8,4,2,7,254,252,76,74
 275 FOR X=0 TO 7: READ y: POKE
USR "j"+x,y: NEXT x: REM J
 280 DATA 0,0,0,4,72,223,62,8
 285 FOR X=0 TO 7: READ y: POKE
USR "k"+x,y: NEXT x: REM K
 290 DATA 0,0,2,7,126,252,76,146
 295 FOR X=0 TO 7: READ y: POKE
USR "p"+x,y: NEXT x: REM P
 300 DATA 224,16,32,124,126,63,1
5,4
 305 FOR x=0 TO 7: READ y: POKE
USR "s"+x,y: NEXT x: REM S
 310 DATA 24,126,255,255,255,126
,24,24
 400 REM Making larger graphics
 ********************************
**
 405 FOR x=0 TO 7: READ y: POKE
USR "f"+x,y: NEXT x: REM F
 410 DATA 16,96,224,160,32,32,48
,56
 415 FOR x=0 TO 7: READ y: POKE
USR "g"+x,y: NEXT x: REM G
 420 DATA 61,63,63,55,51,51,99,9
9
 425 FOR x=0 TO 7: READ y: POKE
USR "l"+x,y: NEXT x: REM L
 430 DATA 0,0,0,0,0,31,63,127
 435 FOR x=0 TO 7: READ y: POKE
USR "m"+x,y: NEXT x: REM M
 440 DATA 0,0,112,240,248,252,25
2,252
 445 FOR x=0 TO 7: READ y: POKE
USR "n"+x,y: NEXT x: REM N
 450 DATA 255,255,255,255,113,96
,96,96
 455 FOR x=0 TO 7: READ y: POKE
USR "o"+x,y: NEXT x: REM O
 460 DATA 222,206,199,195,195,19
8,192,192
 465 FOR x=0 TO 7: READ y: POKE
USR "q"+x,y: NEXT x: REM Q
 470 DATA 120,255,31,15,24,40,40
,24
 475 FOR x=0 TO 7: READ y: POKE
USR "r"+x,y: NEXT x: REM R
 480 DATA 0,244,240,248,100,82,8
1,208
 485 FOR x=0 TO 7: READ y: POKE
USR "t"+x,y: NEXT x: REM T
 490 DATA 24,60,60,126,126,126,2
55,255
 495 FOR x=0 TO 7: READ y: POKE
USR "u"+x,y: NEXT x: REM U
 500 DATA 255,255,255,255,126,60
,24,24
 505 GO SUB 5000
```

I got this from Doug Gillespie. Thanks Doug!



```
  10 FOR a=25 TO 230 STEP 10
  20 CIRCLE a,25,25
  30 NEXT a
  40 FOR a=35 TO 150 STEP 10
  50 CIRCLE 230,a,25
  60 NEXT a
  70 FOR a=220 TO 25 STEP -10
  80 CIRCLE a,150,25
  90 NEXT a
 100 FOR a=150 TO 25 STEP -10
 110 CIRCLE 25,a,25
 120 NEXT a
```

By Doug Gillespie

Take Care All! See you next month!!!